IN THE CLAIMS

1. (previously presented): A multilevel texture processing method for mapping multiple images onto a 3D model with a texture mapping, the method comprising the steps of:

providing an image to the 3D model;

converting the image and the texture mapping to a same spatial coordinate system and dividing them into a plurality of polygons;

extracting overlapped polygons from the image with the texture mapping within the spatial coordinate system;

using the pixel intensity of the overlapped polygons to compute a statistics mean for adjusting the pixel intensity of the image accordingly;

using a prescribed condition to select the texture of one of the image and the texture mapping as the texture of the polygon;

smoothing the texture of the polygon;

making the pixels inside the polygon continuous; and

restoring the polygon and outputting the 3D model.

2. (original): The method of claim 1, wherein the prescribed condition is selected from the group consisting of resolution, polygon orientation, and camera viewing perspective.

3. (original): The method of claim 1, wherein the step of smoothing the texture of the polygon includes texture normalization and texture blurring.

4. (original): The method of claim 3, wherein the texture normalization uses the pixel intensities of the polygons in both the image and the texture mapping to compute a weighted average for adjustment.

5. (original): The method of claim 3, wherein the texture blurring uses the textures of the polygon and its neighboring polygons to compute a weighted average for adjustment.

6. (previously presented): The method of claim 1, wherein the step of making the pixels inside the polygon continuous is achieved by mixing colors with the neighboring polygons.

7. (previously presented): The method of claim 6, wherein the step of mixing colors with the neighboring polygons includes the steps of:

extracting a pixel on the border of the polygon with discontinuous colors; and

computing a weighted average of the intensities of the pixel and its nearest neighboring pixels as a new intensity of the pixel.

8. (previously presented): The method of claim 7, wherein the step of computing a weighted average of the intensities of the pixel and its nearest neighboring pixels as a new intensity of the pixel is followed by the steps of:

computing the difference between the weighted average intensity and the original pixel intensity; and

using the pixel intensity difference to adjust the intensities of the rest of the pixels inside the polygonal texture.

9. (New): A multilevel texture processing method for mapping multiple images onto a 3D model with a texture mapping, the method comprising the steps of:

providing an image to the 3D model;

converting the image and the texture mapping to a common spatial coordinate system and dividing them into a plurality of polygons;

comparing the image with the texture mapping within the spatial coordinate system to extract overlapped polygons;

using the pixel intensity of the overlapped polygons to compute a statistics mean for adjusting the pixel intensity of the image accordingly;

using a prescribed condition to select the texture of one of the image and the texture mapping as the texture of the polygon;

smoothing the texture of the polygon;

making the pixels inside the polygon continuous; and

restoring the polygon and outputting the 3D model,

wherein the pixel intensity of the image is adjusted by a formula:

$$I'_s(x_i, y_i) = I_s(x_i, y_i) - \mu_s + \mu_b,$$

μ_s representing the averaged pixel intensity of the overlapped polygons on the 3D model, μ_b representing the averaged pixel intensity of the overlapped polygons of the input image, $I_s(x_i, y_i)$ representing the pixel intensity of each point on the 3D model, and $I'_s(x_i, y_i)$ representing the adjusted pixel intensity of each point on the 3D model.

10. (New): The method of claim 9, wherein the prescribed condition is selected from the group consisting of resolution, polygon orientation, and camera viewing perspective.

11. (New): The method of claim 9, wherein the step of smoothing the texture of the polygon includes texture normalization and texture blurring.

12. (New): The method of claim 11, wherein the texture normalization uses the pixel intensities of the polygons in both the image and the texture mapping to compute a weighted average for adjustment.

13. (New): The method of claim 12, wherein the weighted average for adjustment is computed by a formula:

$$T'_s(x_i, y_i) = T_s(x_i, y_i) - \mu_s + \mu_b,$$

μ_s representing the averaged pixel intensity of the overlapped polygons on the 3D model, μ_b representing the averaged pixel intensity of the overlapped polygons of the input image, $T_s(x_i, y_i)$ representing the texture pixel intensity at each point in the polygon, and $T'_s(x_i, y_i)$ representing the adjusted texture pixel intensity at each point in the polygon.

14. (New): The method of claim 11, wherein the texture blurring uses the textures of the polygon and its neighboring polygons to compute a weighted average for adjustment.

15. (New): The method of claim 9, wherein the step of making the pixels of the polygon texture continuous is achieved by mixing colors with the neighboring polygons.

16. (New): The method of claim 15, wherein the step of mixing colors includes the steps of:

extracting a pixel on the border of the polygon with discontinuous colors; and

computing a weighted average of the intensities of the pixel and its nearest neighboring pixels as a new intensity of the pixel.

17. (New): The method of claim 16, wherein the step of computing a weighted average of the intensities of the pixel and its neighboring pixels as a new intensity of the pixel is followed by the steps of:

computing the difference between the weighted average intensity and the original pixel intensity; and

using the pixel intensity difference to adjust the intensities of the rest pixels inside the polygonal texture.

18. (New): The method of claim 17, wherein the intensities of the rest pixels inside the polygonal texture is adjusted by a formula:

$$T'_s(x, y) = T_s(x, y) + \sum_{i=1}^{N} w_i \cdot Id_i$$

w_i representing a relevant weight, Id_i representing the pixel intensity difference, N representing the number of total adjustments, $T_s(x_p, y_p)$ representing the pixel intensity of each point on the 3D model, and $T'_s(x_p, y_p)$ is the adjusted pixel intensity of each point on the 3D model.